What is Computer Science? What is CompSci 201?

https://duke.is/cs201-02

Alex Steiger and Owen Astrachan

asteiger@cs.duke.edu, ola@duke.edu

CompSci 201

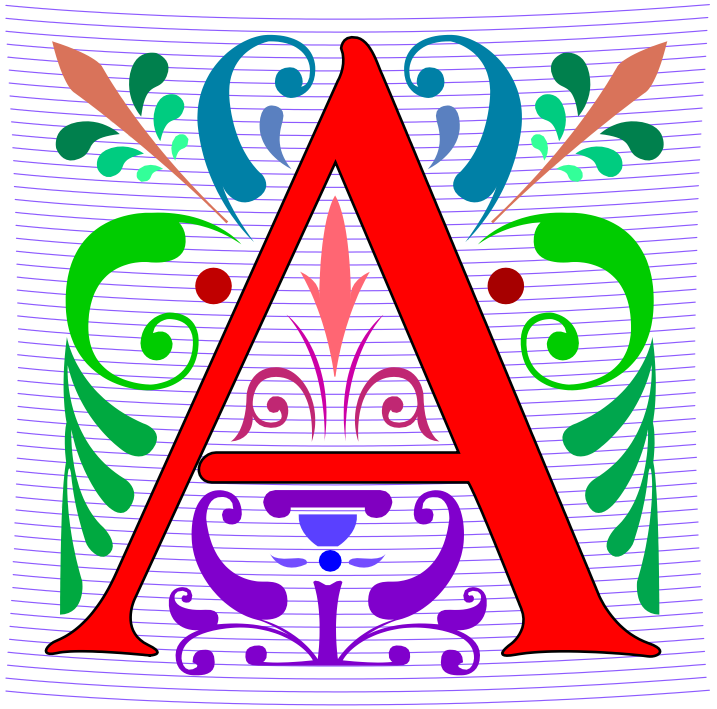
- Everything online

duke.is/cs201-02

- What is WOTO? **WO**rking **TO**gether
  - Key aspect of 201 and compsci@duke

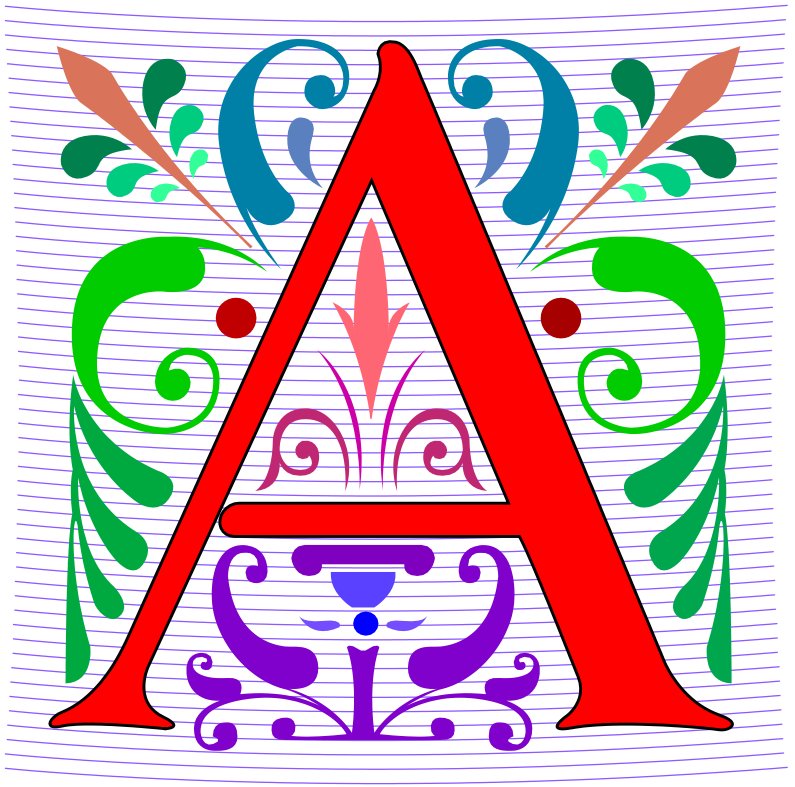
A is for …

- ***Algorithm***
  - Fundamental to Computer Science
- ***API***
  - Libraries are the future



A is for …

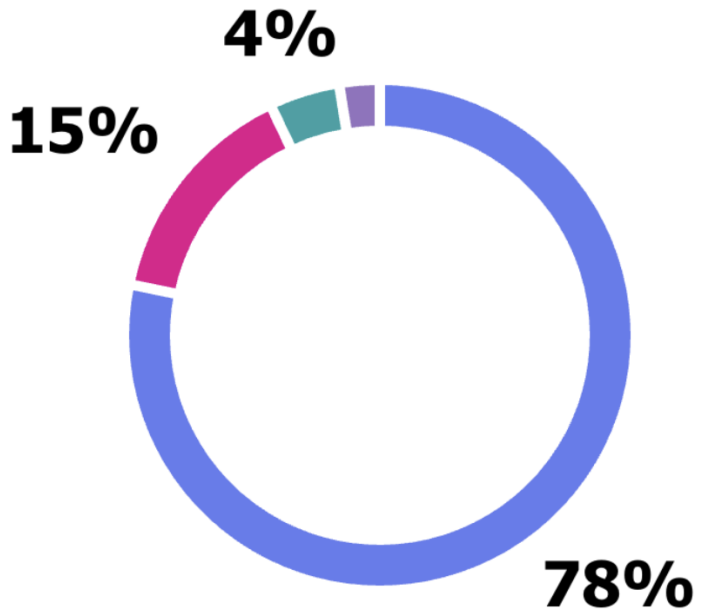
- ***Array***
  - Memory mapped to structure
- ***AI***
  - Machine Learning and …



Who are you?

- First semester at Duke?

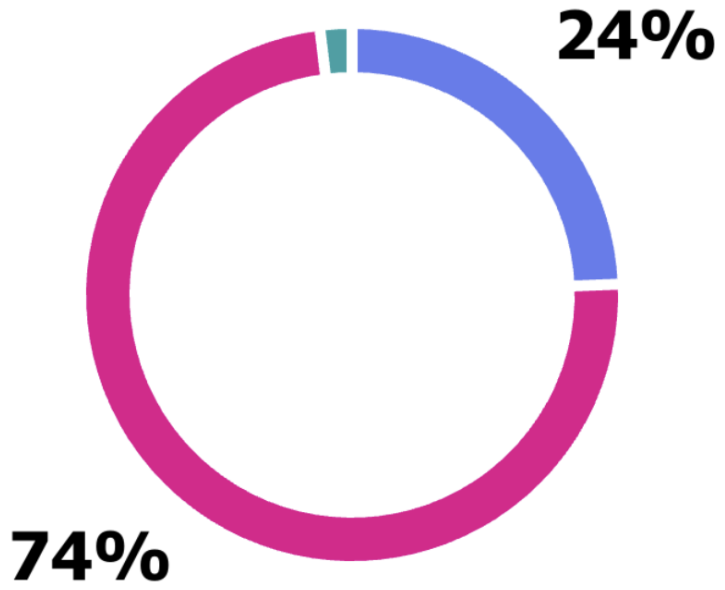
Fall 25	122
Fall 24	23
Fall 23	7
Fall 22	4
Before fall 22	0



Who are you?

- School at Duke?

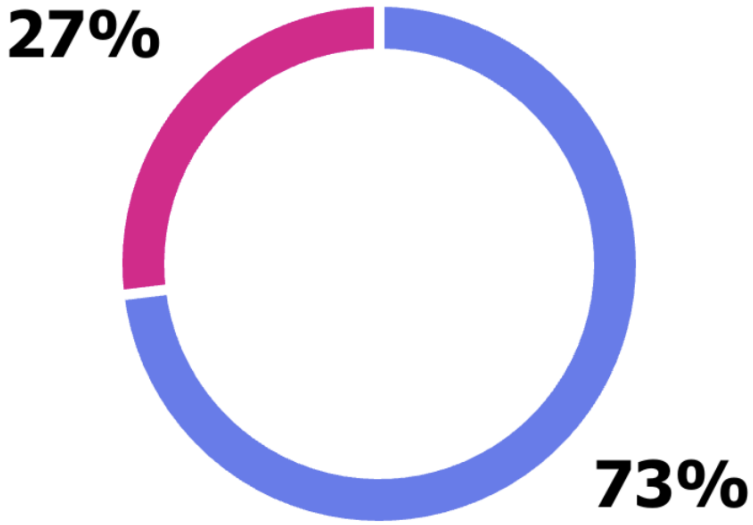
Pratt	38
Trinity	115
Other	3



Who are you?

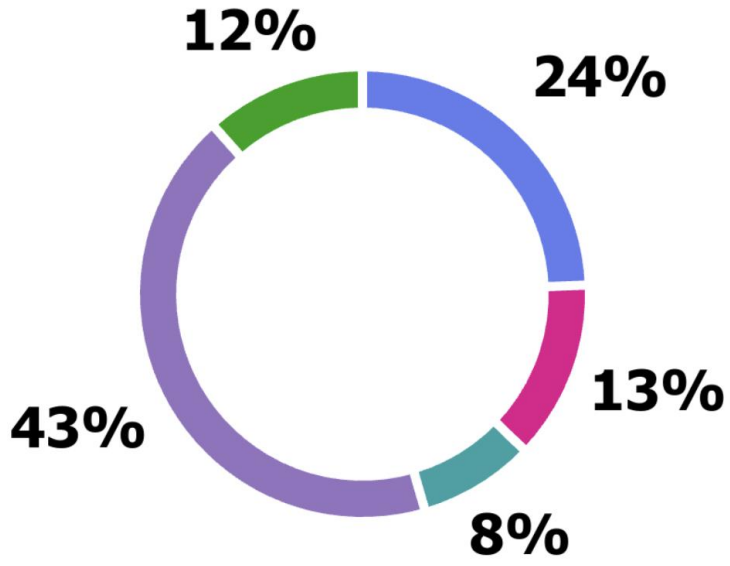
- Laptop of choice?

Mac	114
Windows	42
Linux	0
Do not have a laptop	0

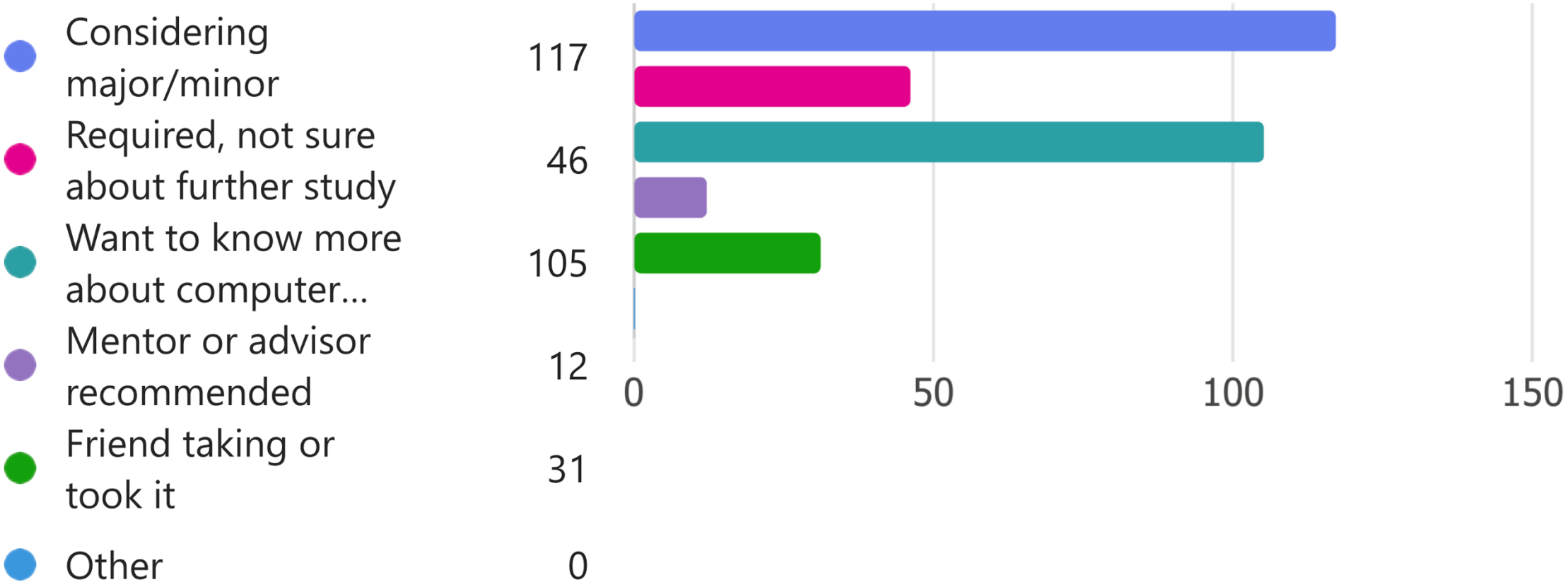


Last programming course I took was...

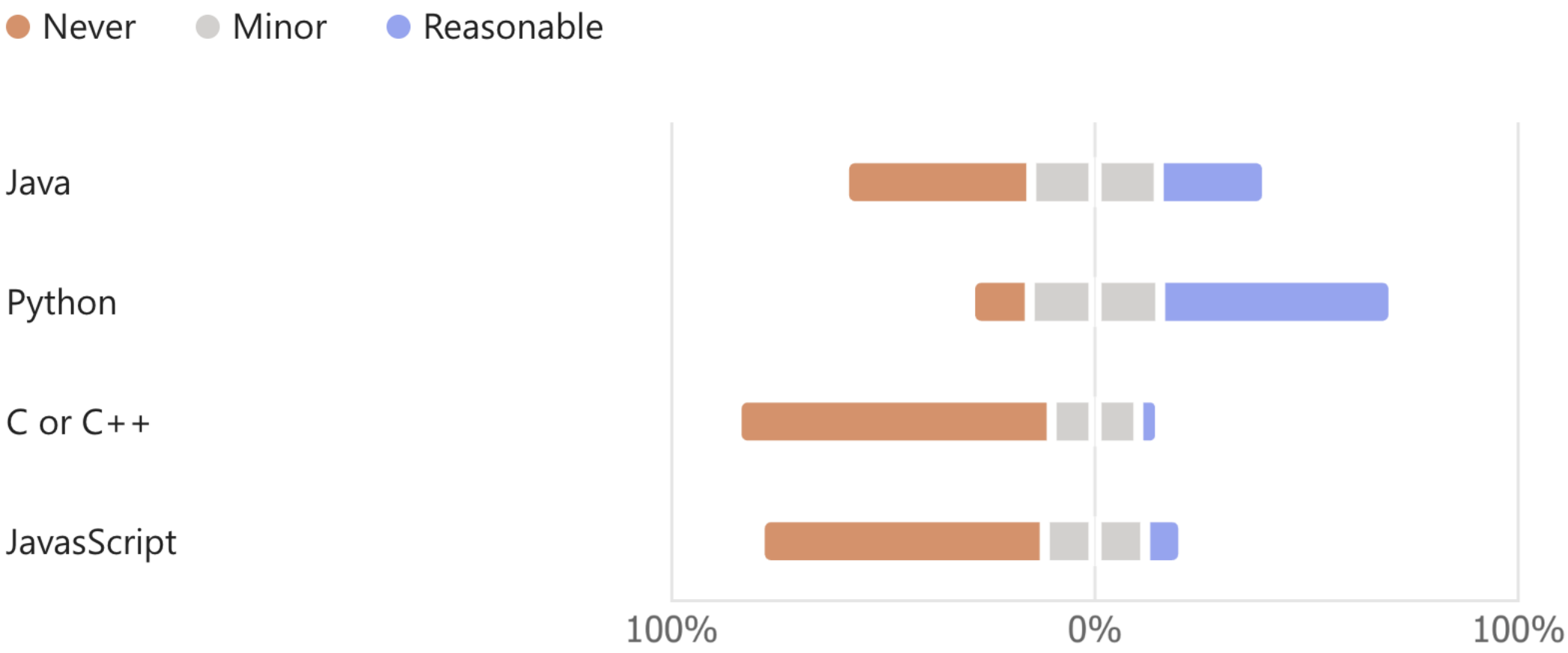
Compsci 101 at Duke	38
EGR 105 (or 103) at Duke	20
Online course	13
Course in high school (AP or other)	67
This is my first programming *course*	18



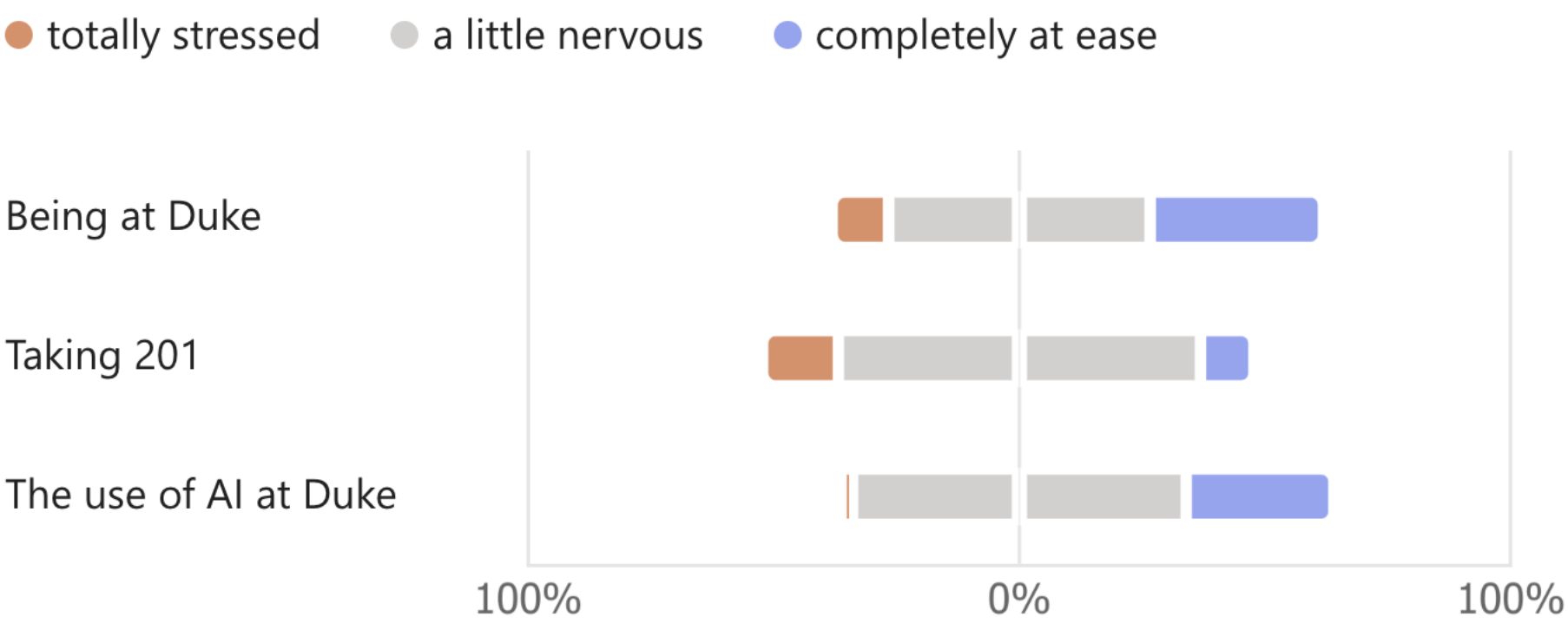
Why am I taking 201?



What language(s) do you know?

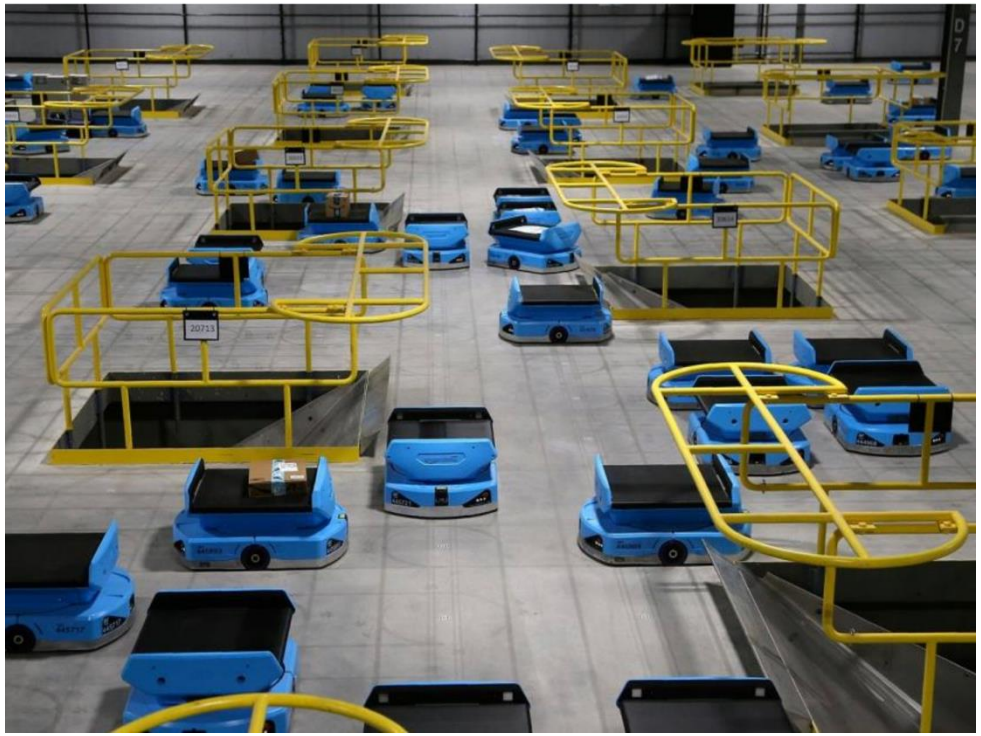


Stress and what to do



Who am I?

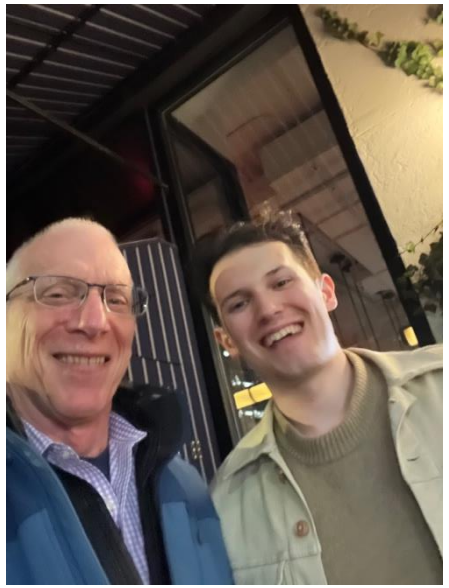
Prof. Alex Steiger

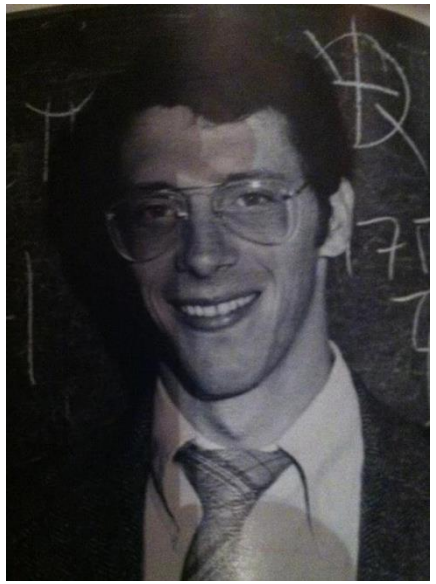


Who am I?

Prof. Astrachan, Owen,
ola, Owen, Prof. Astrachan






What is Computer Science?

"Our species needs, and deserves, a citizenry with minds wide awake and a basic understanding of how the world works."

- Carl Sagan

What is Computer Science?



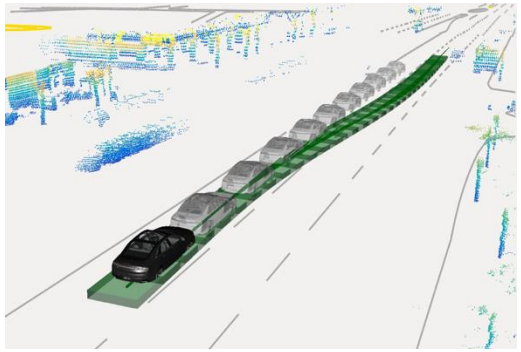
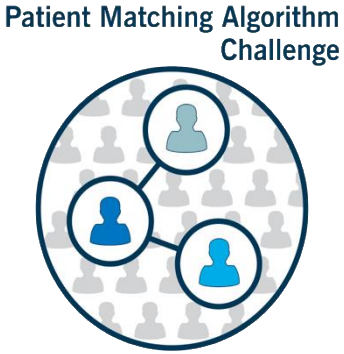



> 5 billion snaps/day; 500 million Insta stories/day

> 14 billion Google queries/day

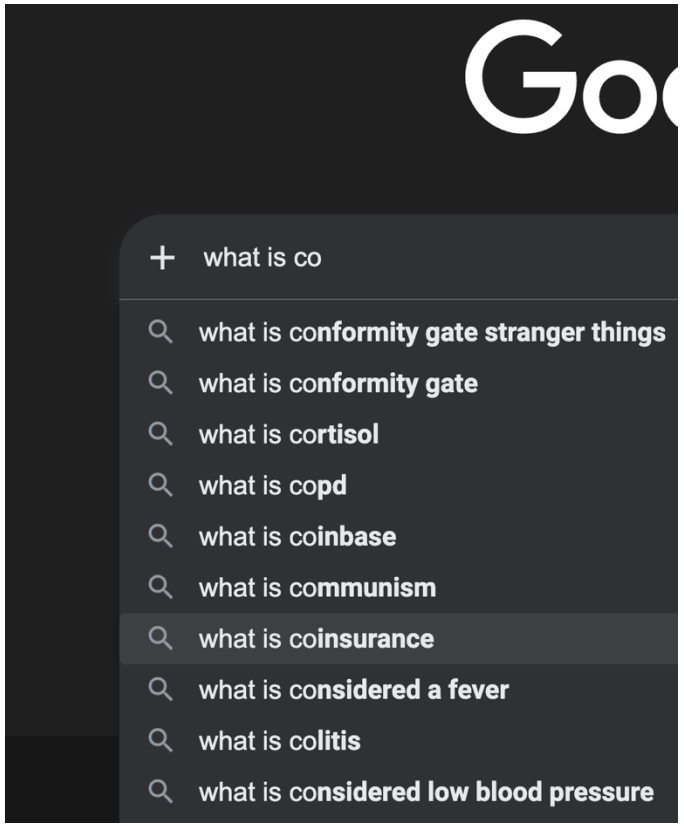
> 2.5 billion ChatGPT queries per day

> one billion TikTok views/day; 93 trillion eyeblinks/day

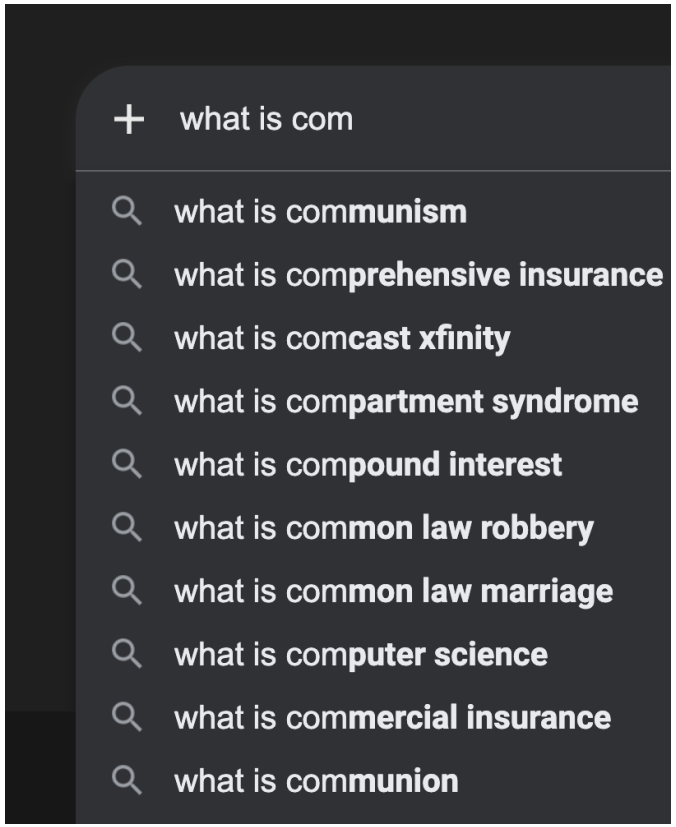




What does Google say …

- January 6, 2026

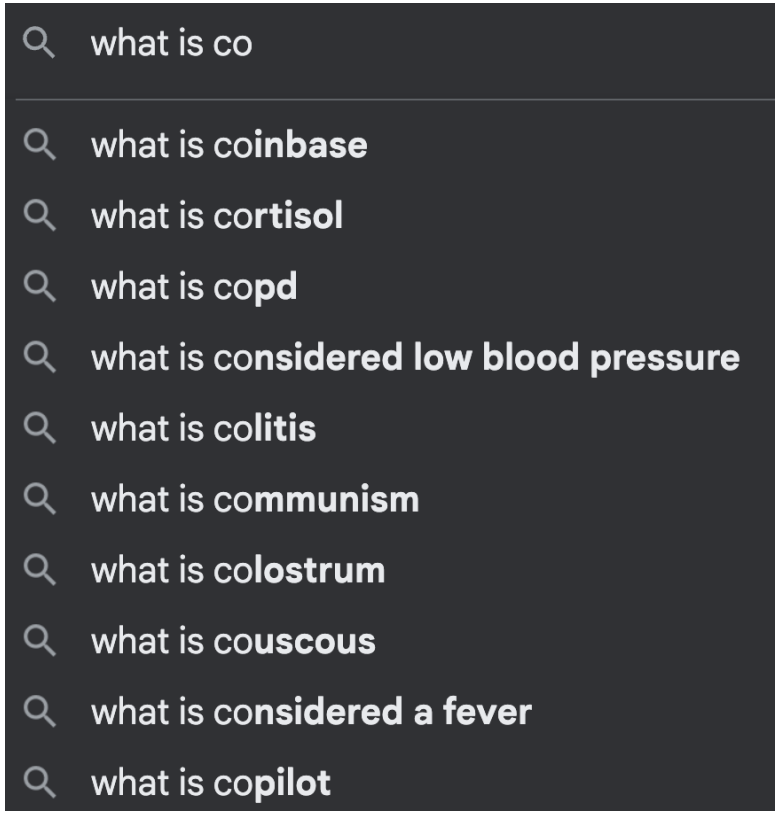


co → com

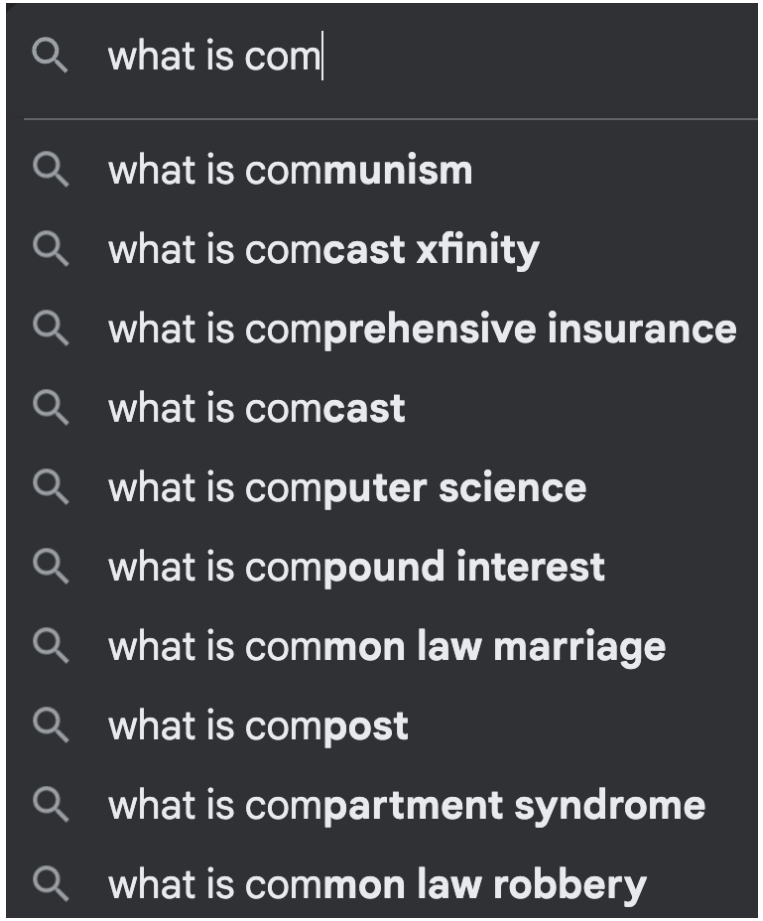


What does Google say… "What is co…"

August 18, 2025



co → com



What did Google say? What is co …

August 18, 2023

August 23, 2024

Add some letters …

AI Overview

Computer science is **the study of computers and computational systems, focusing on algorithms, hardware, software, and their applications to solve problems and automate processes, blending theory (like computation theory) with practical design and implementation (like AI and cybersecurity)**. It's about understanding how information is processed, stored, and communicated, using principles from math, logic, and engineering to create new technologies and systems that impact society.

What is Computer Science?

What is it that distinguishes it from the separate subjects with which it is related?

What is the linking thread which gathers these disparate branches into a single discipline?

Sir (Tony) C.A.R. Hoare, 1971

What is Computer Science?

My answer to these questions is simple—*it is the art of programming a computer.*

Sir (Tony) C.A.R. Hoare, 1971

What is Computer Science?

It is the art of designing efficient and elegant methods of getting a computer to solve problems, theoretical or practical, small or large, simple or complex.

Sir (Tony) C.A.R. Hoare, 1971

What is your definition?

- We will be learning together
  - The we is intentional

- You'll learn more if you're curious and explore, going beyond … to ***engage with material***

  - You'll learn far more than the basics by simply thinking about what you hear

WOTO: https://duke.is/cs201-pl

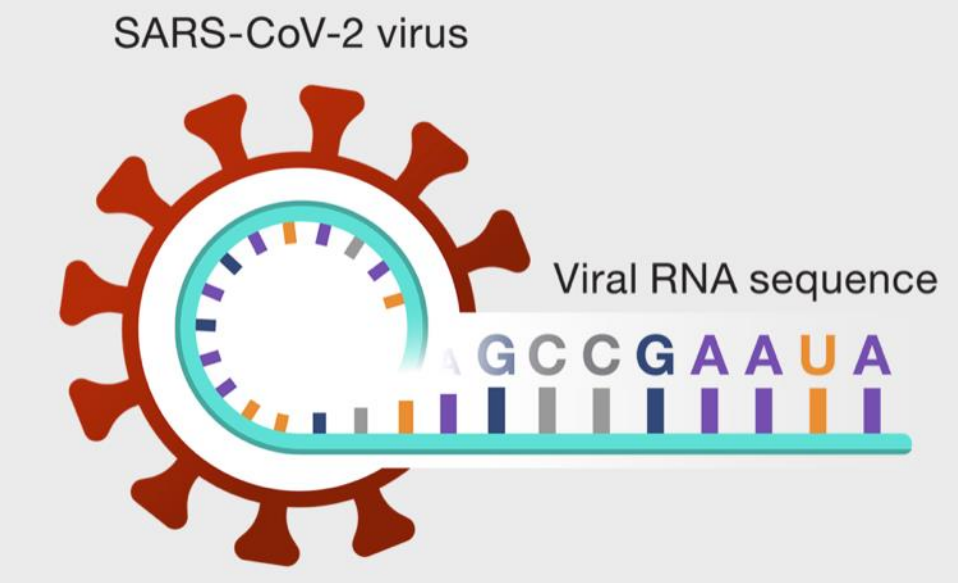
CompSci 201 Partial definition

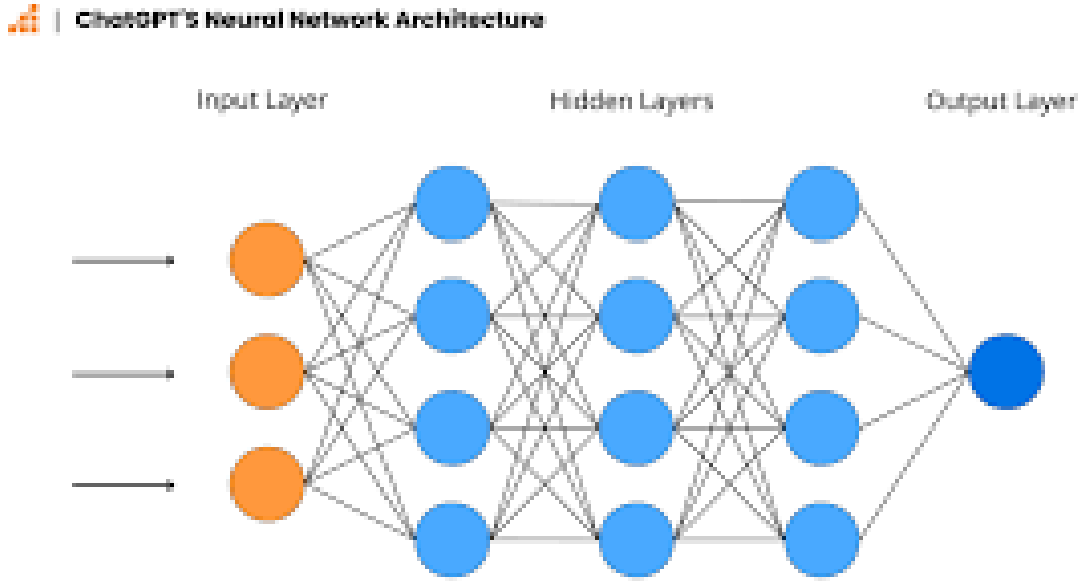
It is the study of automating algorithmic processes that scale

- Multiplying numbers?
- Solving a reCaptcha?
- Sending $$ via Venmo?
- Creating/using ChatGPT?

Impacts of automating at scale

Google





NETFLIX

Who is CS? Latanya Sweeney

"As a Professor of Government and Technology in Residence at Harvard University, my mission is to create and use technology to assess and solve societal, political and governance problems, and to teach others how to do the same.

Former CTO of the FTC, First African-American Women to earn CS PhD from MIT (2001)

Latanya Sweeney

I am a computer scientist with a long history of weaving technology and policy together to remove stakeholder barriers to technology adoption. My focus is on "computational policy" and I term myself a "computer (cross) policy" scientist. I have enjoyed success at creating technology that weaves with policy to resolve real-world technology-privacy clashes.

Her best known academic work is on the theory of *k*-anonymity, and she is credited with the observation that "87% of the U.S. population is uniquely identified by date of birth, gender, postal code".

Famous Computer Scientists (AI Mode '26)

Famous computer scientists include pioneers like **Alan Turing** (theoretical CS, Turing Machine) and **Grace Hopper** (programming languages), innovators like **Tim Berners-Lee** (World Wide Web) and **Linus Torvalds** (Linux), foundational figures such as **Ada Lovelace** (first programmer) and **John von Neumann**, and modern tech leaders like **Bill Gates**, **Larry Page**, and **Mark Zuckerberg**.

From ChatGPT, January 2026

Foundations & Theory

- **Alan Turing** – Computability, Turing machines, WWII codebreaking
- **Alonzo Church** – Lambda calculus, foundations of computation
- **Kurt Gödel** – Logic, limits of formal systems
- **John von Neumann** – Computer architecture, game theory

Algorithms & Data Structures

- **Edsger W. Dijkstra** – Shortest paths, structured programming
- **Donald Knuth** – *The Art of Computer Programming*
- **Robert Tarjan** – SCCs, union–find
- **Niklaus Wirth** – Pascal, Modula

Programming Languages & Software Engineering

- **Grace Hopper** – Compilers, COBOL
- **Barbara Liskov** – Liskov Substitution Principle
- **Bjarne Stroustrup** – C++
- **James Gosling** – Java

Artificial Intelligence & Machine Learning

- **John McCarthy** – Coined "AI," Lisp
- **Marvin Minsky** – Early AI, cognitive models
- **Geoffrey Hinton** – Neural networks
- **Yann LeCun** – CNNs

Computer Scientists … famous?

August 18, 2025


Margaret Hamilton


Brian Kernighan


Donald Knuth


James Gosling


Barbara Liskov


Charles Babbage


Larry Page


Shafi Goldwasser


Bill Gates


Frances Elizabeth ...


Guido van Rossum


J. C. R. Licklider

This list is biased toward …

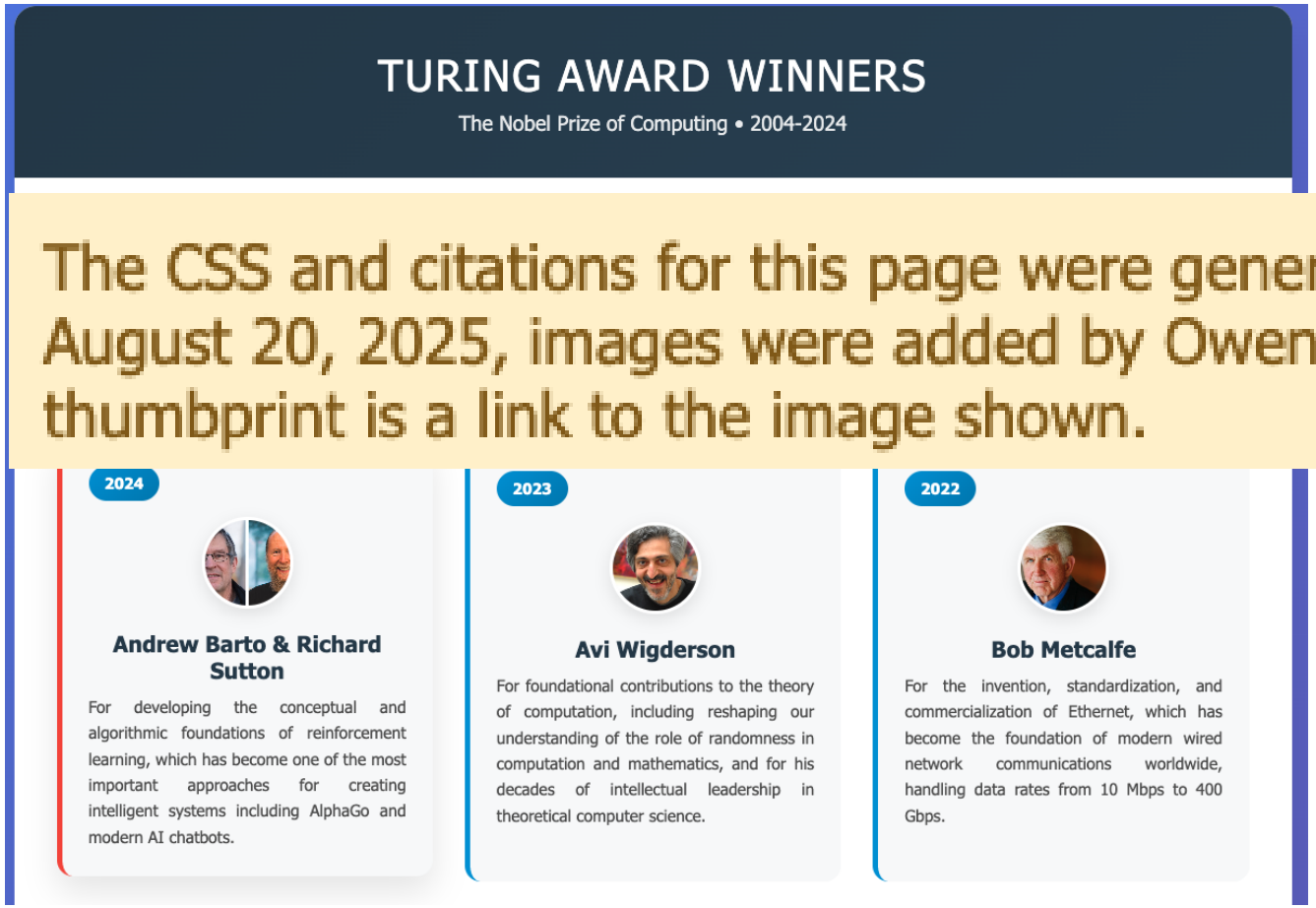
1. **Alan Turing**
2. **Donald Knuth**
3. **John McCarthy**
4. **Edsger W. Dijkstra**
5. **Ada Lovelace**
6. **Barbara Liskov**
11. **Vinton Cerf**
12. **Brian Kernighan**
13. **Ken Thompson**
14. **Leslie Lamport**
15. **Geoffrey Hinton**
16. **Yoshua Bengio**

ChatGPT, 2025

Alan Turing, Ada Lovelace, John von Neumann, Donald Knuth, Alan Kay, Grace Hopper, Tim Berners-Lee, Linus Torvalds, Dennis Ritchie, Ken Thompson, Brian Kernighan, Edsger Dijkstra, Tony Hoare, Barbara Liskov, Frances Allen, Shafi Goldwasser, Leslie Lamport, Geoffrey Hinton, Yann LeCun, Andrew Ng, and Judea Pearl.

From Claude

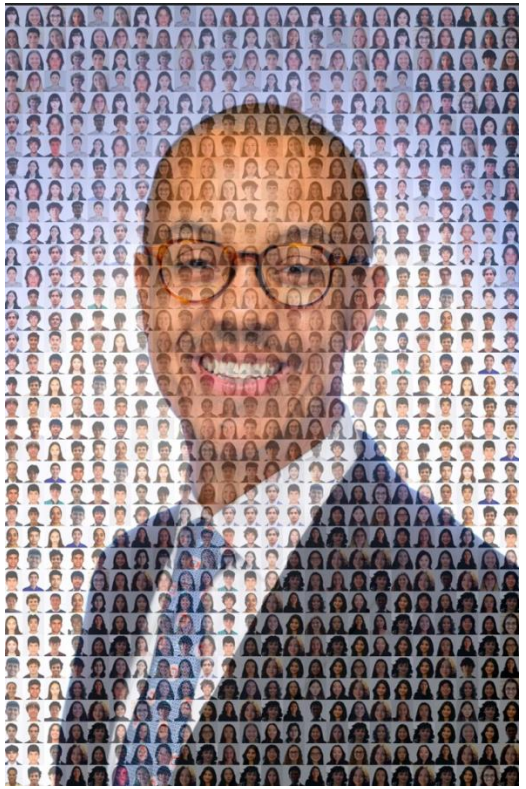
Turing Award Winners https://bit.ly/201-turing



Computer Scientists!

More algorithmically

https://bit.ly/gb-201-2026



More algorithmically

https://bit.ly/jl-342-spring26

Why Computer Science at Duke?

LinkedIn Top Colleges 2025: The 50 best colleges for long-term career success in the U.S.

LinkedIn News
18,603,802 followers

August 12, 2025

🏭 **Top industries:** Technology & Internet, Financial Services, Business Consulting & Services | 📍 **Top locations:** New York City, Washington, D.C., Raleigh, N.C. | 🧠 **Most notable skills:** MATLAB, Web Development, AI Engineering

Let's talk about programming

Learning Java is part of 201

- Many of you have studied Java
- Many of you have never studied Java

- This course is about data structures and algorithms, realized in Java

- Is it a programming course? Yes and No

Class Resources

- See duke.is/cs201-02 and course syllabus for details on what's accessible

- Use Ed accessed from Canvas
- Use ChatGPT or other LLMs to understand

- We will study Java, but it's a quick intro

Learning Goals I

Given a problem statement and a real data source, design, develop, debug, and test a Java program that uses appropriate standard libraries to efficiently solve the problem.

- Realize this goal in the course projects and APTs that are fundamental to CompSci 201

Learning Goals II

Write programs that effectively implement and use data structures such as: arrays, maps, linked lists, stacks, queues, trees, and graphs.

- This will include java.util classes, 201 DIY classes, and some you implement yourself.
- Realize this goal in course projects and APTs

Learning Goals III

Evaluate the time and space complexity of iterative and recursively-defined algorithms using empirical and mathematical analysis.

- Compsci 201 is ***fundamentally about tradeoffs***, this requires reasoning, measuring, thinking
- Realize this goal by developing code for projects, answering questions for those, and for WOTOs

Learning Goals IV

Utilize LLMs to explore data structures and algorithms, to help develop solutions to problems without sacrificing learning and understanding, and to understand basic Java concepts

- Realize this goal in responding to WOTO and programming for projects and APTs

Course Goal, not a learning goal

- Develop connections among your peers, at Duke, with course staff
- Understand computer science as interdisciplinary, experience that joy
- WOrking TOgether

First Program in Java

- Use VSCode to execute and study program
- Read a text-file, count how many words there are and how many unique words there are
  - Measure how long the program takes, later examine runtime analytically
- You may not understand the Java, but you'll hopefully understand the structure of the code

Where is class code?

- Gitlab repo: https://bit.ly/201-spring26-git

What does this code do?

```
1  import java.util.*;
2  import java.io.*;
3
4  public class StaticUniqueWords {
       Run | Debug
5      public static void main(String[] args) throws Exception {
6          Scanner s = ne
7          ArrayList<Stri
8
9          double start =
10         while (s.hasNe
11             String wor
12             list.add(w
13         }
14         double end = S
15         double time =
16         System.out.pri
17         System.out.printf("time: %2.3g\n", time);
18         s.close();
19     }
20 }
```

```
double start = System.nanoTime();
while (s.hasNext()) {
    String word = s.next().toLowerCase();
    list.add(word);
}
double end = System.nanoTime();
```

Use libraries, declare variables

```
1  import java.util.*;
2  import java.io.*;
3
4  public class StaticUniqueWords {
       Run | Debug
5      public static void main(String[] args) throws Exception {
6          Scanner s = new Scanner(new File("data/kjv10.txt"));
7          ArrayList<String> list = new ArrayList<>();
```

Access libraries

What is a file?

What is an ArrayList?

What does new do?

Understanding Repetition

- When does loop terminate? What takes time?

```
Scanner s = new Scanner(new File("data/kjv10.txt"));
ArrayList<String> list = new ArrayList<>();

double start = System.nanoTime();
while (s.hasNext()) {
    String word = s.next().toLowerCase();
    list.add(word);
}
double end = System.nanoTime();
double time = (end-start)/1e9;
```

.hasNext and .next working together

what is a double?

Understanding methods/functions

```
double start = System.nanoTime();
while (s.hasNext()) {
    String word = s.next().toLowerCase();
    list.add(word);
}
double end = System.nanoTime();
```

Toward understanding Java

- The "dot" . invokes a method/function
  - Sometimes passing a parameter: **`list.add(word)`**
  - Typically returning a value: **`s.next()`**

- Typically, on an *object*: String or Scanner
- Sometimes on a *class*: System

- Terms object and class discussed next class

Modify to track unique words

```
 9  double start = System.nanoTime();
10  while (s.hasNext()) {
11      String word = s.next().toLowerCase();
12      if (! list.contains(word)) {
13          list.add(word);
14      }
15  }
16  double end = System.nanoTime();
```

"not" in Java is !

Equivalent in Python

```
4   f = open(fName)
5   words = []
6   count = 0
7   start = time.perf_counter_ns()
8   for line in f:
9       data = line.lower().split()
10      for w in data:
11          count += 1
12          if w not in words:
13              words.append(w)
14  end = time.perf_counter_ns()
15  elapsed = (end-start)/1e9
```

Analyze Code, Algorithm, Structures

- What file of a million strings results in fastest execution? Slowest?
  - How do we find code bottlenecks?
  - How do we experiment with code?

```
12      if (! list.contains(word)) {
13          list.add(word);
14      }
```

`list.contains` dissected

- What type of value is returned by method?
  - See how it's used in code: part of if statement
- What type of value is the parameter?
  - Here: **`String`**, in general: **`ArrayList<Type>`**
- Code within the method?
  - Loop over all values looking for parameter

Lots of Java/Language details! But …

- This code/algorithm ***does not scale***!
  - `list.add(..)`  `list.contains(..)` 

- Some Java concepts familiar from previous programming: Python or other language
  - You'll learn vocabulary and practice!

High-level Java Types

- Java is an object-oriented language
  - Classes are templates or object factories,
  - Call new to create object from class template

- Primitive types: int, double, char, boolean, …
  - Fast, small, values stored in memory
  - All other types are references to objects

Primitive types seen in code

- double: floating point values
  - `System.nanoTime(), (end-start)/1e9`
- int: integer values
  - implicit in `list.size()`
- boolean: true/false
  - `s.hasNext()`

Object or Reference types seen in code

- String: sequence of characters
  - returned by **`s.next()`**
  - returned by **`xyz.toLowerCase()`**

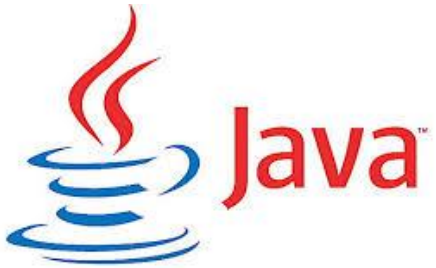
- java.io.File — references file "on disk"
- java.util.Scanner — facilitates reading file
- java.util.ArrayList — stores object references

High-level Java Collections

- Arrays are homogeneous collections
  - Like Python lists, Matlab array/vector
  - Once created, cannot grow, ***can hold primitives***

- ArrayList<>, Set<>, Map<> are collections
  - Dynamic, powerful, efficient at scale
  - ***Cannot store primitives***: Integer 😊 int 😢

WOTO: https://duke.is/cs201-pl



th +

1 out of 454

pro +

Course Policies and Details

- See website and syllabus for complete info
- Midterms, Final
  - Take best of midterm and corresponding final
- Help, collaboration, individual assessments
- Time, late policies, academic integrity

- Use and misuse of ChatGPT and other LLMs

Course Details: see syllabus

Activity	Hours/Week	Days of Week
Class/WOTO	1.25 * 2 = 2.5	Two Days
Engage (read/think)	1 x 2 = 2.0	scattered
Discussion	1.5	Fri
APT	varies, 2-4	scattered
Projects	varies 2-6	scattered

Grading Details

Category	Percentage
Assignments	26%
Three Midterms and Final	44%
APTs	8%
APT Quizzes	8%
Engagement Points	7%
Discussion	7%

Progressively challenging
Weights 1, 2, 3

Three@10%, final 14%
max(mid,final)
no help

Weekly, help, but
"On your own"

No help allowed

Engagement Points: need 300/450

- WOTO, on time questions during class
  - Up to 200 points
- Several other ways to engage
  - Programming, reading, thinking, project early
- Discussion, assignments, more, …

DukeGPT, ChatGPT, LLMs

- Learn, be curious, answer questions
- Yes: generate ideas, Avoid: generate code

- **G**row (code) Gradually
- **T**est (code) Thoroughly
- **H**ack (code) Humbly and Honestly
- **C**ode Collaboratively

GTHC

Success in Computer Science 201 is …

WOTO

Working Together

https://201wo.to